IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 4, 2018, accepted November 30, 2018, date of publication December 13, 2018,
date of current version January 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2885070

A Security-Enhanced Cluster Size Adjustment Scheme for Cognitive Radio Networks

ANITA LATSMI MANOHAR[1], KOK-LIM ALVIN YAU[1], (Senior Member, IEEE), MEE HONG LING[1], AND SULEMAN KHAN[2], (Member, IEEE)

[1]Department of Computing and Information Systems, Sunway University, Subang Jaya 47500, Malaysia
[2]School of Information Technology, Monash University Malaysia Campus, Subang Jaya 47500, Malaysia

Corresponding author: Kok-Lim Alvin Yau (koklimy@sunway.edu.my)

ABSTRACT Cognitive radio network (CRN) is the next generation wireless network that allows unlicensed users [secondary users (SUs)] to explore and use the underutilised licensed channels (white spaces) owned by licensed users (primary users). The purpose is to increase the spectrum utilization for enhanced network performance. Clustering segregates SUs in a CRN into logical groups (clusters) with each consisting of a leader (cluster head) and member nodes. A budget-based cluster size adjustment scheme is applied to enable each cluster to adjust its number of member nodes in its cluster based on the availability of white spaces in order to improve network scalability. However, cluster size adjustment is prone to attacks by malicious SUs that launch random and intelligent attacks. Hence, we incorporate an artificial intelligence approach called reinforcement learning (RL) into a trust model to countermeasure the random and intelligent attacks. The simulation results show that RL-based trust model increases the utilization of white spaces and cluster size to improve network scalability and enhance network performance despite the presence of RL-based intelligent attacks.

INDEX TERMS Artificial intelligence, reinforcement learning, attacks, trust model, cognitive radio.

I. INTRODUCTION

Cognitive radio (CR) enables unlicensed users (or secondary users, SUs) to explore and use underutilised licensed channels (or white spaces) owned by licensed users (or primary users, PUs) in order to increase the overall spectrum utilisation and network performance [1]. A distributed CR network (CRN) consists of a number of SUs who communicate among themselves in the absence of fixed network infrastructure, such as an access point or a base station [2]. Clustering segregates SUs in a CRN into logical groups (or clusters). Each cluster consists of a leader (or cluster head) and member nodes as shown in Figure 1. A SU member node can be a single or multiple hops away from the SU cluster head [3]. This means that a SU member node – being a child – can connect to a cluster head – being a parent – directly, or via an upstream member node. The cluster heads and some of their member nodes form a backbone route leading to a base station. Larger cluster size improves network scalability, and it offers two main advantages. Firstly, the overhead reduces since: a) routing messages are exchanged among nodes in a backbone route only, and b) clustering messages for updating the network changes (e.g., the demand for white spaces) are exchanged at the cluster level only rather than the network level [4]. The problem is that malicious nodes can launch random or intelligent attacks against cluster size, impeding the goals of clustering to achieve the two main advantages. Intelligent attacks enable malicious nodes to adapt their attack strategies in order to prevent themselves from being identified. Hence, in this research, an artificial intelligence approach called reinforcement learning (RL) is incorporated into a trust model to countermeasure the random and intelligent attacks. The intelligent trust models enable cluster heads to adapt to the dynamicity of attack strategies in order to identify malicious nodes. This is a challenging problem because, not only does RL being incorporated in the trust model of a SU cluster head, it is incorporated in the attack model of a malicious SU node.

The rest of this section presents overviews of clustering and cluster size adjustment. Subsequently, the security issues of cluster size adjustment are presented to provide further description on the problem of achieving the right cluster size in order to achieve the two main advantages of clustering under the random and intelligent attacks launched by malicious nodes. Next, RL is presented. Finally our contributions are presented. The rest of this paper is organized as follows.

TABLE 1. General, clustering and RL notations used in this paper .

Category	Notation	Description
General	$t \in T$	A decision epoch (e.g., time window) with $T = \{1, 2, \ldots, t, \ldots\}$
	$m \in M$	A SU m with a set of SUs $M = \{1, 2, \ldots, m, \ldots, \lvert M\rvert\}$
	Γ_m	A set of SU neighbour nodes Γ_m of SU m
	$j \in J$	A PU j with a set of PUs $J = \{1, 2, \ldots, j, \ldots, \lvert J\rvert\}$. PU j's activity is either on or off in channel j
	λ_j^{ON}, λ_j^{OFF}	ON and OFF rates of channel j, respectively
	T_j^{ON}, T_j^{OFF}	ON and OFF periods of channel j, respectively
	$P_{j,t}^{ON}$, $P_{j,t}^{OFF}$	Probabilities of channel j being ON and OFF at decision epoch t, respectively
Clustering	$C_c \in C$	A cluster C_c with a set of clusters being $C = \{C_1, C_2, \ldots, C_c, \ldots, C_{\lvert C\rvert}\}$
	$CH_c \in CH$	A cluster head CH_c of cluster C_c with a set of cluster heads $CH = \{CH_1, CH_2, \ldots, CH_c, \ldots, CH_{\lvert C\rvert}\}$
	$MN_{c,m} \in MN_c$	A set of member nodes of cluster in the network, $MN_{c,m} = \{MN_{c,1}, MN_{c,2}, \ldots, MN_{c,m}, \ldots, MN_{c,\lvert C\rvert}\}$
	$\mu_{m,c}$	Clustering message sent by SU m in cluster C_c
	$nodeState_m$	Node state of SU m with $nodeState_m \in (CH, MN, NC)$. CH represents cluster head, MN represents member node, and NC represents nonclustered node
	$C_{c,m}$	A set of neighbouring clusters of SU m with $C_{c,m} = \{C_{m,1}, C_{m,2}, \ldots, C_{c,m}, \ldots, C_{\lvert C,M\rvert}\}$
	T_{scan}	Channel sensing interval of a channel for receiving clustering message
	$\tau_{c,m}$	A token from cluster head CH_c to nonclustered SU m
	$\beta_{c,t}$	Budget value of cluster C_c at decision epoch t
	$N_{c,t}$	Number of tokens available at cluster head CH_c at decision epoch t
	J_c	Number of available common channels in cluster C_c
	D_{J_c}	Threshold for the minimum number of available common channels in a cluster C_c
	$\texttt{TREQ}_{m,c}$	A token request message sent by nonclustered SU m to cluster head CH_c
	$\texttt{TACC}_{c,m}$	A token accept message sent by cluster head CH_c to nonclustered SU m
	$\texttt{TDEC}_{c,m}$	A token decline message sent by cluster head CH_c to nonclustered SU m
	η_{need}	Number of tokens needed by a SU based on hop count
	η_{req}	Number of tokens requested by a SU to cluster head
	η_{given}	Number of tokens given by a cluster head to SU
	η_{waste}	Number of tokens wasted by a SU
RL	$s_{m,t} \in S$	State of SU m at decision epoch t with S being a set of possible states
	$a_{m,t} \in A$	Action of SU m at decision epoch t with A being a set of possible actions
	$r_{m,t+1}(s_{m,t})$	Reward of a state-action pair $(s_{m,t}, a_{m,t})$ of SU m who has taken action $a_{m,t}$ under state $s_{m,t}$ at decision epoch t+1
	$Q_{m,t+1}(s_{m,t}, a_{m,t})$	Q-value of SU m, who has taken action $a_{m,t}$ under state $s_{m,t}$ at decision epoch t, at decision epoch $t+1$
	$\pi^*(s_{m,t})$	Optimal policy that provides a series of actions under different states $s_{m,t}$
	$V^{\pi^*}(s_{m,t})$	Cummulative reward or value function for taking a series of optimal action under different states $s_{m,t}$
	α	Learning rate
	γ	Discount rate

Section II presents related work. Section III presents system model. Section IV presents the RL-based attack model and our proposed RL-based trust model. Section V presents performance evaluation. Table 1 summarizes the general, clustering, and RL notations used in this paper. For simplicity, *SU cluster head* and *SU member node* are referred to as *cluster head* and *member node* henceforth.

A. CLUSTERING

The motivation behind clustering is to achieve three main advantages. Firstly, it improves network scalability. The cluster heads and gateway nodes, which can hear from more than a single cluster and provide inter-cluster communication, form a backbone route leading to the base station. The member nodes only exchange messages with their respective cluster heads via intra-cluster communication. Since routing messages can be exchanged among nodes in a backbone route only, the amount of overhead reduces in clustered networks as opposed to the need to propagate routing messages among all nodes in nonclustered networks. Secondly, it increases cluster stability. Clustering messages for updating the network changes (e.g., the demand for white spaces, channel availability, and network topology) are only exchanged locally for reconfiguration among a cluster head and its member nodes at the cluster level in clustered networks as opposed to the need to propagate updates among all nodes at the network level in nonclustered networks. Thirdly, it facilitates cooperative tasks as the cluster head and its member nodes use a common channel for intra-cluster communication [5]. For instance, in cooperative channel sensing, a cluster head and its member nodes perform channel sensing in a collaborative manner to provide more accurate channel sensing outcomes.

In [6], a clustering scheme forms and maintains clusters based on a mobility model (i.e., zone-based group mobility model) to improve network scalability and cluster stability, as well as to reduce energy consumption in mobile ad-hoc networks. Spectrum-aware clustering schemes take channel availabilities (e.g., the number of available channels with white spaces) into consideration. In [7], a clustering scheme selects cluster heads based on node degree (or the number of neighboring nodes) in the available channels to improve network scalability in a CRN, which has a multi-channel environment. In [8], a clustering scheme forms and maintains clusters based on the number of available channels and the speed of the nodes, as well as the interference level from PUs to improve network scalability, as well as to reduce clustering messages, packet loss, and the number of disconnected nodes, in CRNs. In [9], a clustering scheme: a) determines potential nodes of a route between a source node and a destination node, b) selects cluster heads based on node degree, the number of available channels, remaining energy, the distance to the destination, and the speed of the potential nodes, among the potential nodes, and c) selects member nodes and a common channel for each cluster based on the number of available channels. The clustering scheme has been shown to increase the number of available common channels in a cluster, as well as reduce clustering messages and energy consumption. In [10], a clustering scheme forms clusters based on the number of available channels, remaining energy, and the speed of the nodes, to reduce end-to-end delay performance.

In addition, recent reviews of clustering schemes [5], [11] present various cluster formation and maintenance mechanisms to achieve network scalability and cluster stability, as well as to facilitate cooperative tasks in CR-based networks. At the time this paper is being written, there is only a little effort to investigate cluster size adjustment, which is the focus of this paper.

B. CLUSTER SIZE ADJUSTMENT

Budget-based cluster size adjustment scheme has been proposed in wireless networks [12], although there has only

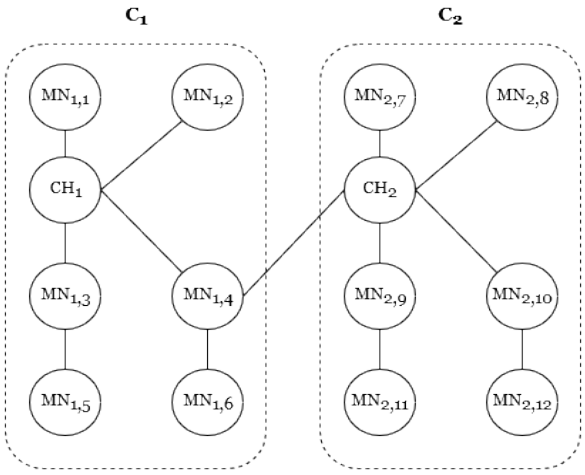

FIGURE 1. An example of an attack scenario in cluster size adjustment. CH_c represents a cluster head CH_c, and $MN_{c,m}$ represents a member node m of cluster C_c. A malicious SU $MN_{1,4}$ joins more than a single cluster.

been perfunctory effort to investigate this approach in CRNs. The cluster head determines the budget value that represents the amount of white spaces at the cluster head, which is dynamic in nature. Subsequently, the cluster head distributes the budget value, in the form of tokens, to neighboring SUs. Each neighboring SU requests for the right number of tokens based on its number of children from the cluster head so that the SU neighbour node, together with its children, become member nodes of the cluster. For instance, in Figure 1, a SU located two hops away from the cluster head (i.e., $MN_{1,5}$) would request two tokens: one is required to send a packet to an upstream parent node (i.e., $MN_{1,3}$), while another one is required to send the packet from the upstream parent node to the cluster head (i.e., CH_1). Hence, with respect to the cluster head, a single-hop SU requests a single token, a two-hop SU requests two tokens, and so on. In CRNs, cluster size adjustment enables cluster heads to adjust their respective cluster sizes (i.e., the number of member nodes in a cluster) according to the availability of white spaces in an adaptive manner. Cluster size adjustment increases the cluster size of a cluster when the amount of white spaces increases, allowing more SUs to join the cluster, hence improving network scalability. In this research, cluster size adjustment is first investigated in the presence of malicious attacks.

Ideally, each SU joins one of its neighboring clusters and becomes its member node, or forms its own cluster and becomes a cluster head when none of the neighboring clusters has sufficient tokens (or white spaces) to cater for the SU. The cluster head determines the budget value that represents the amount of white spaces at the cluster head. The cluster head then distributes the budget value, in the form of tokens, to the neighboring SUs, depending on the number of hops between a cluster head and a member node. Therefore, the dynamic budget value causes the cluster head to add or drop nodes accordingly. Due to the dynamic availability of white spaces, the SU member nodes constantly and cooperatively exchange clustering messages (e.g., the demands for white spaces) with their respective cluster heads in order to maintain an ideal cluster size.

C. SECURITY ISSUES OF CLUSTER SIZE ADJUSTMENT

Unfortunately, malicious SU member nodes tend to launch attacks in two ways to reduce the utilisation of white spaces in order to reduce network scalability. Firstly, a malicious SU member node joins more than a single cluster to increase its availability of white spaces. This means that a malicious SU member node, who has joined a single cluster, continues to join another cluster so that it can request white spaces (or tokens) from the two clusters, depriving legitimate SUs from joining the cluster and accessing the white spaces. Secondly, a malicious SU member node requests more white spaces (or tokens) from its cluster head(s). The cluster heads of the two clusters may not be aware that a malicious SU member node has joined more than a single cluster due to the lack of communication and cooperation with each other for two reasons. Firstly, the clusters may operate in different channels in a multi-channel environment. Secondly, the cluster heads may be located out of each other's transmission range. This causes the cluster head to allocate more white spaces than the right amount, which again deprives legitimate SUs of accessing the white spaces. As a consequence, cluster size becomes smaller, causing network scalability to reduce. Moreover, the occurrence of a node joining or leaving a cluster increases clustering overhead. The malicious SU member nodes can launch two types of attack strategies: a) *random attacks* in which attacks are launched in a random manner; and b) *intelligent attacks* in which attacks are launched using RL to maximize its detrimental effects while avoiding being detected [13]. The goal of the legitimate SUs is to form the right cluster size in which the member nodes of a cluster make full use of the budget value, and hence increase the utilisation of white spaces leading to improved network scalability. To the best of our knowledge, the investigation on the effects of RL-based attacks to cluster size adjustment is first of its kinds.

Figure 1 shows an example of an attack scenario in cluster size adjustment. There are two clusters, namely C_1 and C_2. The cluster C_1 consists of a cluster head CH_1 and SU member nodes $MN_{1,1}$, $MN_{1,2}$, $MN_{1,3}$, $MN_{1,4}$, $MN_{1,5}$ and $MN_{1,6}$. The cluster C_2 consists of a cluster head CH_2 and SU member nodes $MN_{2,7}$, $MN_{2,8}$, $MN_{2,9}$ $MN_{2,10}$, $MN_{2,11}$ and $MN_{2,12}$. Consider: a) cluster C_1 operates in channel $j = 1$ and cluster C_2 operates in channel $j = 2$, and b) cluster heads CH_1 and CH_2 are located out of each other's transmission range. Suppose, SU $MN_{1,4}$ joins both clusters C_1 and C_2, and requests for tokens from cluster heads CH_1 and CH_2. It requests $\tau_{1,4} = 1$ token from cluster head CH_1 and $\tau_{2,4} = 1$ token from cluster head CH_2, and enjoys more transmission opportunities. The malicious nature of such node reduces network scalability as legitimate nodes may not join the cluster due to insufficient available tokens. Cluster heads CH_1

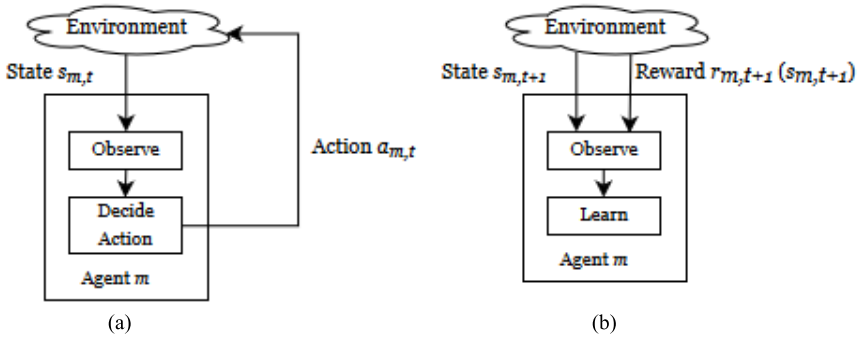

FIGURE 2. A RL agent m in its operating environment at decision epochs t and t+1. (a) Desition epoch t. (b) Desition epoch $t+1$.

and CH_2 may not be aware of $MN_{1,4}$ joining their clusters simultaneously as they are operating in different channels. Therefore, the goal of the malicious node $MN_{1,4}$ is to join more than a single cluster so that it is allocated with more tokens, and hence more white spaces (or transmission opportunities). This deprives legitimate SUs of accessing the white spaces.

By requesting more tokens than necessary, malicious SU member nodes can manipulate the budget value, and hence the cluster size. In this paper, the budget value represents the amount of white spaces at a cluster head, and so attacks can reduce the utilisation of white spaces. With different representations for the budget value, the attack scenario can change. Consider a cooperative channel sensing scheme in CRNs. The cluster head performs decision fusion on channel sensing outcomes received from member nodes to produce final decisions on channel availability. The budget value can be based on the accuracy of the final decisions, whereby a larger cluster size increases the number of channel sensing outcomes received from member nodes, contributing to a better accuracy. Malicious nodes can launch attacks to reduce cluster size, and hence the accuracy of the final decisions, resulting in increased interference to PUs' activities.

D. REINFORCEMENT LEARNING

Reinforcement learning (RL), which is an unsupervised artificial intelligence approach [14], is embedded in agents or decision makers to observe, learn, and select the optimal action under the current operating environment for performance enhancement as time goes by. Without learning, an agent must use a predefined set of rules that may not cater for all kinds of operating environment encountered throughout an agent's operation, causing suboptimal network performance. Learning is inevitable as the optimal action varies in a dynamic operating environment. RL has been widely applied in wireless networks, particularly CRNs, to provide network performance enhancement [15], and its application to security enhancement has become popular [13]. As an example, RL is embedded in a *SU cluster head* to observe, learn, and identify malicious SU nodes that launch random and intelligent attacks in order to improve network performance. The use of RL is necessary because of the dynamicity of the attack strategy adopted by the malicious SU nodes. Nevertheless, the use of RL has increased security vulnerabilities due to the need to observe and learn from the operating environment which can be manipulated. As an example, RL is embedded in a malicious SU node who intelligently adjusts its attack strategy to maximize the detrimental effects while switching between legitimate and malicious behaviors to avoid being detected. In [2], it has been shown that, intelligent attacks can cause an agent to fail to achieve convergence to optimal action. Due to the popularity of the use of RL in networking schemes, and the lack of focus on intelligent attacks using RL, this paper focuses on both aspects and investigates them in the context of cluster size adjustment, which has received little focus in the literature. The RL model has three main representations, namely state, action, and delayed reward. The agent observes the *state*, and selects an *action* to maximize the *delayed reward*, which represents the performance metrics. Q-learning is a popular RL approach [14] that estimates the Q-values $Q_{m,t}(s_{m,t}, a_{m,t})$ of a state-action pair updated using the Q-learning function as follows:

$$Q_{m,t+1}(s_{m,t}, a_{m,t}) \leftarrow (1-\alpha)Q_{m,t}(s_{m,t}, a_{m,t}) + \alpha[r_{m,t+1}(s_{m,t+1}) + \gamma \max_{a \in A} Q_{m,t}(s_{m,t+1}, a)] \quad (1)$$

Based on Figure 2 and Algorithm 1, an agent m observes state $s_{m,t}$. The agent then selects either an exploration or an exploitation action. Using ε-greedy approach, a random action (or an exploration action) $a_{m,t}$ is selected with a small probability ε to learn the Q-value of the action in order to discover the best-known action, while the best-known action, $a_{m,t}$ with the highest Q-value (or the exploitation action) is selected with probability $(1-\varepsilon)$. The agent m then observes state $s_{m,t+1}$. For each state-action pair, an agent m observes its delayed reward $r_{m,t+1}(s_{m,t+1})$, which is a short-term reward received at decision epoch $t+1$ after taking an action $a_{m,t}$

Algorithm 1 Q-Learning Algorithm at Agent m

1: Repeat
2: Observe state $s_{m,t}$
3: Determine exploration or exploitation
   i. If exploration, choose a random action $a_{m,t}$
   ii. If exploitation, choose the best known action $a_{m,t}$ using Equation (2)
4: Observe state $s_{m,t+1}$
5: Receive delayed reward $r_{m,t+1}(s_{m,t})$
6: Update Q-value $Q_{m,t+1}(s_{m,t}, a_{m,t})$ using Equation (1)

under state $s_{m,t}$ at decision epoch t. The learning rate α determines the extent to which the newly acquired knowledge overrides the previously learned Q-value $Q_{m,t}(s_{m,t}, a_{m,t})$. The discounted reward $\gamma \max_{a \in A} Q_{m,t}(s_{m,t+1}, a)$ represents the cumulative rewards received by the agent m at decision epoch $t = 1, 2, \ldots$. An agent m learns the optimal policy $\pi^*(s_{m,t})$ that provides a series of optimal actions under different states in order to maximize the cumulative reward $V^{\pi^*}(s_{m,t})$, or value function, as follows [14]:

$$\pi^*(s_{m,t}) = \underset{a \in A}{\operatorname{argmax}}\, Q_{m,t}(s_{m,t}, a) \tag{2}$$

$$V^{\pi^*}(s_{m,t}) = \max_{a \in A} Q_{m,t}(s_{m,t}, a) \tag{3}$$

E. OUR CONTRIBUTIONS

The contribution of our research is to address the presence of malicious nodes that launch random and intelligent attacks against cluster size adjustment. We address this critical issue by using a RL-based trust model, which is a framework that assigns trust value to each SU member node based on its actions in order to identify and withdraw malicious SU member nodes that join more than a single cluster or request more tokens than required from their cluster head(s). While the RL model can be embedded in a cluster head to implement the trust model, it can also be embedded in a malicious node to launch intelligent attacks. A legitimate (malicious) SU member node has a higher (lower) trust value. To the best of our knowledge, the investigation to countermeasure intelligent attacks against cluster size adjustment in CRNs is first of its kind.

II. RELATED WORK

This section presents cluster size adjustment, attacks in CRNs, and RL-based Trust Model.

A. CLUSTER SIZE ADJUSTMENT

While general reviews on clustering and cluster size adjustment schemes have been investigated in wireless sensor networks (WSNs) [16]–[21], mobile ad hoc networks [22], there has only been perfunctory effort to investigate cluster size adjustment schemes in CRNs.

In [23] and [24], a biclique graph approach is proposed to enable a node to construct two types of graphs, namely bipartite and biclique. A bipartite graph shows the relationship of different characteristics (e.g., the number of available common channels in a cluster and the number of member nodes in a cluster), while a biclique graph shows characteristics (e.g., cluster size) adjusted based on a set of rules. As an example, using the rule a + b, the biclique graph maximizes the number of member nodes a, and the number of available common channels b, in a cluster.

In [25], a membership reassignment approach is proposed to adjust the cluster size based on two strategies so that the clusters in the network have higher local uniformity (or lower variation in the number of nodes in a cluster). Firstly, if the cluster size of a cluster is less than a cluster size threshold, a physically closest member node from a neighboring cluster whose size is greater than the cluster size threshold switches its membership to join the cluster in order to reduce the number of single-node and small clusters. Secondly, each cluster invites physically closest neighbour nodes to join its cluster until the cluster size threshold is reached in order to reduce variation in the number of nodes among the clusters. In [26], another membership reassignment approach is proposed to enable a base station to receive information from the clusters, and reassign the membership of nodes who can hear from more than a single cluster (e.g., gateway nodes) in order to adjust the cluster size of the clusters so that all clusters in a network have approximately the same cluster size.

In [27], cluster size is restricted to a maximum number of hops between a cluster head and each of its member node in order to improve network scalability in WSNs, although there is non-uniform resource availability (e.g, channel capacity) among the nodes. In [4], SpectruM-Aware clusteR-based rouTing (SMART) scheme is proposed to enable SUs to adjust the number of nodes in a cluster and search for a route on a clustered network. In [28], a cluster size adjustment scheme based on the distance between the sensor nodes and base stations is proposed to improve energy utilisation of the sensor nodes. In [29], a cluster size adjustment scheme based on RL is proposed to improve network scalability and cluster stability in CRNs. A budget-based approach is used to ensure that the number of nodes in a cluster is restricted to a budget value [30], [31]. In [30], a budget-based approach is proposed to reduce the number of isolated nodes so that such nodes can join a cluster even if the token has run out in order to further improve network scalability. In [31], two budget-based approaches are proposed, namely *Rapid* and *Persistent*. Consider a tree structure rooted at the cluster head. In *Rapid*, tokens are propagated downstream in a one-way direction, so surplus tokens cannot be reused resulting in smaller cluster size. In *Persistent*, tokens are propagated downstream in a two-way direction, so surplus tokens can be redistributed by parent nodes resulting in larger cluster size.

This paper extends the budget-based cluster size adjustment scheme with a trust model to address malicious attacks. Our proposed trust model can adapt to the dynamicity of attack strategies adopted by the malicious nodes. This is a challenging issue because an attack strategy may change

as time goes by, so malicious nodes can be difficult to be identified.

B. ATTACKS IN CRNS

While various kinds of attacks, including PU emulation attacks [32]–[34], spectrum sensing data falsification (SSDF) attacks [35]–[37], byzantine attacks [38], [39], unintentional attacks [40], [41], random attacks [42], [43], bias attacks [42], [43], and denial of service (or jamming) attacks [32], [44], [45], [46] have been well investigated in CRNs, there is lack of investigation on intelligent attacks against cluster size adjustment schemes. In [47], intelligent malicious nodes launch jamming attacks against the cluster head whenever traffic presents at the cluster head, causing the cluster head, as well as its neighboring cluster heads, to become malicious, deteriorating the intra-clusters and inter-clusters communication. In [41], a malicious SU launches intelligent attacks by relying on the legitimate SUs sensing outcomes, and only launch attacks when all the honest SUs provide the same sensing outcomes, hence the attacker provides the opposite sensing outcome. In [48], an intelligent malicious node launches sybil attacks that enables nodes to switch between multiple identities, which enhances its impact of attack without being detected and enhances its performance using RL.

To the best of our knowledge, the investigation on random attacks and intelligent attacks to cluster size adjustment, as well as the investigation on RL-based trust model to address such attacks, is first of its kinds. By applying RL, both malicious, as well as legitimate SUs that turn malicious, can be detected as time goes by.

C. RL-BASED TRUST MODEL

RL has been applied to trust models to identify malicious nodes in CRNs. In [35], the malicious nodes send inaccurate sensing outcomes to neighboring nodes or a fusion center in order to manipulate final decision on channel availability in SSDF attacks. In [48], the malicious nodes launch sybil attacks by generating multiple false sensing reports to a fusion center in spectrum leasing. In [41], trust and reputation management has been applied to detect malicious SUs, as well as legitimate SUs that turn malicious, in CRNs.

To the best of our knowledge, the investigation on RL-based trust model to countermeasure intelligent attacks in cluster size adjustment is first of its kinds. This is a challenging issue because we are using reinforcement learning to tackle intelligent attacks.

III. SYSTEM MODEL

This section presents the system model, covering the local operating environment, cluster formation and node joining.

A. LOCAL OPERATING ENVIRONMENT AND WHITE SPACES

The CRN is comprised of SU $m \in M = \{1, 2, \ldots, |M|\}$. Each PU $j \in J = \{1, 2, \ldots, |J|\}$ transmits in its own channel j, and so there are $|J|$ available channels. A decision epoch t consists of time windows with $\{1, 2, ..., \vartheta\}$ timeslots. A PU $j's$ activity follows a Poisson ON-OFF model. The ON state represents that a PU j occupies a channel j in which the ON period is T_j^{ON}, and the OFF state represents otherwise in which the OFF period is T_j^{OFF}. The ON-OFF periods are exponentially distributed with the ON and OFF arrival rates λ_j^{ON} and λ_j^{OFF}, respectively [49], [50]. At decision epoch t, the probabilities of channel j being ON and OFF are as follows [51]:

$$P_{j,t}^{ON} = \frac{\lambda_j^{OFF}}{\lambda_j^{ON} + \lambda_j^{OFF}} - \frac{\lambda_j^{OFF}}{\lambda_j^{ON} + \lambda_j^{OFF}} e^{-(\lambda_j^{ON} + \lambda_j^{OFF})t} \quad (4)$$

$$P_{j,t}^{OFF} = \frac{\lambda_j^{ON}}{\lambda_j^{ON} + \lambda_j^{OFF}} + \frac{\lambda_j^{OFF}}{\lambda_j^{ON} + \lambda_j^{OFF}} e^{-(\lambda_j^{ON} + \lambda_j^{OFF})t} \quad (5)$$

where higher probability $P_{j,t}^{OFF}$ indicates a higher amount of white spaces in channel j at decision epoch t.

B. CLUSTER FORMATION

In a cluster formation procedure, each cluster $C_c \in C = \{C_1, C_2, \ldots, C_{|C|}\}$ consists of a cluster head CH_c and member nodes $MN_{c,m} \in MN_c$. Upon initialization, each SU m is nonclustered $nodeState_m = NC$. Each SU m senses for each available channel $j \in J$ for a time interval T_{scan}, and joins one of the neighboring clusters $C_c \in C_{c,m} \subseteq C$, and becomes its member node $MN_{c,m}$. Each cluster C_c must possess at least a single common channel $J_c \geqslant 1$ to facilitate intra-cluster communication. Member node $MN_{c,m}$ sends clustering message $\mu_{m,c}$ in its cluster C_c.

There are two main cases for a nonclustered SU m. Firstly, a SU m does not receive any clustering messages $\mu_{m,c}$, so it forms its own cluster C_c and becomes a cluster head CH_c. Secondly, a SU m receives at least a single clustering message from multiple neighboring clusters $|C_{c,m}| > 1$. If there are no suitable cluster head CH_c, then the SU m becomes a cluster head itself. Upon becoming a cluster head, the cluster head CH_c calculates the budget value of its cluster $\beta_{c,t}$ based on the dynamic amount of white spaces available at the cluster head at decision epoch t. Specifically, the budget value is $\beta_{c,t} = P_{j,t}^{OFF} \times \vartheta$. For instance, the probability of a channel j being OFF at decision epoch t is $P_{j,t}^{OFF} = 0.6$, and each time window has 15 time slots, so the budget value is $\beta_{c,t} = P_{j,t}^{OFF} \times \vartheta = 0.6 \times 15 = 9$, whereby the maximum number of available tokens in the cluster C_c is given by $N_{c,t} = \beta_{c,t} = 9$. Subsequently, the cluster head distributes the budget value to its neighboring nodes Γ_m in the form of tokens $\tau_{c,m \in \Gamma_m}$. Specifically, the budget value of a cluster $\beta_{c,t}$ changes from time to time with respect to the PUs' activities at each decision epoch t. The cluster head CH_c then distributes $N_{c,t}$ available tokens at each decision epoch t.

C. NODE JOINING AND LEAVING

The node joining procedure for a nonclustered SU node to join a cluster after receiving a token from a SU cluster head

Algorithm 2 Cluster Formation and Maintenance at Nonclustered SU m

*/*Part I: Scan each channel for clustering message*/*
while J **do**
  scan channel $j \in J$ for T_{scan}
  if receive $\mu_{m,c}$ from C_c **then**
    store $\mu_{m,c}$
  end if
end while
*/*Part II: Send and process received clustering message */*
while not receive $\texttt{TACC}_{c,m}$ **do**
  send $\texttt{TREQ}_{m,c}$ to C_c
  if receive $\texttt{TACC}_{c,m}$ **then**
    store $\tau_{c,m}$
    $nodeState_m = MN$
  else receive $\texttt{TDEC}_{c,m}$
  end if
end while

Algorithm 3 Cluster Formation and Maintenance at SU Cluster Head CH_c

*/*Part I: Scan common channel of cluster to update budget*/*
while T **do**
  scan common channel j for decision epoch t
  update $\beta_{c,t} = P_{j,t}^{OFF} \times \vartheta$
  update $N_{c,t} = \beta_{c,t} - N_{c,t-1}$
end while
*/*Part II: Send and process tokens*/*
receive $\texttt{TREQ}_{m,c}$ from nonclustered SU m
if $N_{c,t} \geqslant 1$ and $J_c \geqslant D_{J_c}$ **then**
  $N_{c,t} = N_{c,t}$ - $\tau_{c,m}$
  send $\texttt{TACC}_{c,m}$ with $\tau_{c,m}$ to SU m
else
  send $\texttt{TDEC}_{c,m}$ to SU m
end if
*/*Part III: Withdraw member nodes when budget is insufficient*/*
if $N_{c,t} < 0$ **then**
  send $\texttt{TDEC}_{c,m}$ to $MN_{c,m}$
end if

is shown in Algorithm 2. The node joining and leaving procedures for a SU cluster head is shown in Algorithm 3. The node leaving procedure due to insufficient budget for a SU member node is shown in Algorithm 4.

In *Part I* of Algorithm 2, a nonclustered SU m scans each available channel $j \in J$ for a time interval T_{scan} to receive and store clustering message $\mu_{m,c}$ sent by SU neighboring nodes Γ_m. In *Part II* of Algorithm 2, SU m sends and receives clustering message $\mu_{m,c}$ (e.g., token request message $\texttt{TREQ}_{m,c}$ and token accept message $\texttt{TACC}_{c,m}$). The SU m sends a $\texttt{TREQ}_{m,c}$ to a cluster head CH_c in cluster C_c. The SU m receives a token accept message $\texttt{TACC}_{c,m}$ from cluster head CH_c, stores the token $\tau_{c,m}$, and becomes its member node $MN_{c,m}$. Note that, the nonclustered SU m joins a cluster based on the decisions made by cluster heads, and so they do not consider the criteria for node joining. If a SU m fails to join any clusters (e.g., none of the neighboring clusters has sufficient number of available tokens), it becomes a cluster head itself with $nodestate_m = CH$.

In *Part I* of Algorithm 3, a SU cluster head CH_c scans its common channel j for a decision epoch t and updates: a) the budget value $\beta_{c,t}$ of the cluster C_c, which is equivalent to the amount of white spaces in the common channel j at each decision epoch t, specifically $\beta_{c,t} = P_{j,t}^{OFF} \times \vartheta$, and b) the number of tokens available $N_{c,t}$ at the cluster C_c based on the updated budget value $\beta_{c,t}$ at each decision epoch t, specifically $N_{c,t} = \beta_{c,t} - N_{c,t-1}$, where $N_{c,t-1}$ is the number of tokens available in the cluster before the update. Both $\beta_{c,t}$ and $N_{c,t}$ are updated from time to time to maintain an ideal cluster size. In *Part II* of Algorithm 3, the criteria for node joining is applied by the SU cluster head CH_c to grant a token $\tau_{c,m}$ to a nonclustered SU m. Upon receiving a token request message $\texttt{TREQ}_{m,c}$ from a nonclustered SU m, the SU cluster head CH_c updates the number of tokens available $N_{c,t}$ in its cluster C_c, and sends a token $\tau_{c,m}$ via a token accept message $\texttt{TACC}_{c,m}$ to the nonclustered SU m if: a) at least a single token is available $N_{c,t} \geqslant 1$, and b) the number of common channels in a cluster upon node joining is greater than its preset threshold $J_c \geqslant D_{J_c}$. Otherwise, the SU cluster head CH_c sends a token decline message $\texttt{TDEC}_{c,m}$ to the nonclustered SU m. In *Part III* of Algorithm 3, a SU cluster head CH_c sends a token decline message $\texttt{TDEC}_{c,m}$ to a SU member node $MN_{c,m}$ if the number of tokens available at the cluster head CH_c at decision epoch t is insufficient, specifically $N_{c,t} < 0$. This allows cluster maintenance whereby the cluster size changes with the budget value, which changes with the amount of white spaces in the common channel, in order to maintain an ideal cluster size as time goes by.

Algorithm 4 Node Leaving From Cluster C_c at SU Member Node $MN_{c,m}$

if receive $\texttt{TDEC}_{c,m}$ **then**
  $nodeState_m = NC$
end if

In Algorithm 4, a SU member node $MN_{c,m}$ leaves its cluster C_c and becomes a nonclustered node when it receives a token decline message $\texttt{TDEC}_{c,m}$ from its cluster head CH_c. This happens when the budget value becomes insufficient, specifically the number of tokens available at the cluster head CH_c at decision epoch t is insufficient $N_{c,t} < 0$. Upon becoming a nonclustered SU m, it undergoes Algorithm 2 to join another cluster with sufficient number of available tokens.

TABLE 2. RL model embedded in a malicious SU *m* for intelligent attacks.

RL element	Representation	Description
State	$s_{m,t} = \frac{\eta_{given}}{\eta_{req}}$ $0 \leq s^1_{m,t} \leq 0.2$ $0.2 < s^2_{m,t} \leq 0.4$ $0.4 < s^3_{m,t} \leq 0.6$ $0.6 < s^4_{m,t} \leq 0.8$ $0.8 < s^5_{m,t} \leq 1$	state $0 \leq s_{m,t} \leq 1$ represents the ratio of the number of tokens given by the cluster head to the number of tokens requested by the malicious SU m, where $s_{m,t} = 0$ and $s_{m,t} = 1$ represent the worst and the best perceptions of the malicious SU m from the cluster head, respectively.
Action	$a_{m,t} = (a_{m_1,t}, a_{m_2,t}) \in A$, $a_{m_1,t} = \{0.1, 0.2, \ldots, 0.9\}$, $a_{m_2,t} = \{0.1, 0.2, \ldots, 0.9\}$	action $a_{m_1,t}$ represents the probability of an attack and $a_{m_2,t}$ represents the intensity of an attack.
Reward	$r_{m,t+1}(s_{m,t+1})$ $=\eta_{waste} = \eta_{req} - \eta_{need}$	reward represents a successful attack, where the amount of wasted white spaces is the difference between the number of tokens requested from the cluster head and the number of tokens needed by the malicious SUs, specifically $\eta_{waste} = \eta_{req} - \eta_{need}$

IV. ATTACK MODEL AND PROPOSED RL-BASED TRUST MODEL

This section presents the attack model adopted by the malicious SUs, and the proposed RL-based trust model.

A. ATTACK MODEL

There are two types of attacks, namely random and intelligent attacks. In random attacks, malicious SUs launch attacks in a random manner without learning about the legitimate SUs' information in the hope of lowering the reward of legitimate SUs. In the intelligent attacks, RL is applied to maximize the effect of the attacks and minimize the chance of being detected by cluster heads. A successful attack maximizes the malicious SUs' reward, however if the malicious SU is detected, it can be expelled from the cluster by a cluster head. So, a malicious SU must choose its attack strategy based on the response from the cluster head. For example, a cluster head reduces the tokens allocated to a malicious SU, giving a reduced reward to the malicious SU. Hence, the malicious SU must adapt its attack strategy so that it is not expelled from the cluster head. Table 2 represents the RL model embedded in a malicious SU for intelligent attacks.

A malicious SU observes the perception it receives from the cluster head (or state), which refers to the number of tokens given by the cluster head η_{given} as opposed to the number of tokens requested to the cluster head η_{req}, specifically $\frac{\eta_{given}}{\eta_{req}}$. The cluster head reduces the distribution of tokens requested by a member node under two circumstances: a) if it detects a malicious SU that launches intelligent attacks and b) if there is insufficient white spaces because the amount of white spaces fluctuates due to the PUs' activities. The malicious SU then takes an action which consists of a combination of the probability of an attack, $0 \leq a_{m_1,t} \leq 1$ and the intensity of an attack, $0 \leq a_{m_2,t} \leq 1$. When a state occurs at decision epoch t, an action is selected to match with the state in which the reward can be optimized. The reward represents a successful attack, or the amount of wasted white spaces $\eta_{waste} = \eta_{req} - \eta_{need}$, where the number of wasted white spaces η_{waste} is the difference between the number of tokens requested from the cluster head η_{req} and the number of tokens needed η_{need} by the malicious SUs. Therefore, when a malicious SU successfully joins more than a single cluster or request for more tokens than required, it wastes the tokens that deprives legitimate SUs of joining the cluster and requesting for tokens. A malicious SU that receives a good perception from the cluster head (i.e., cluster head grants the number of requested tokens) tend to launch attacks with higher probability and greater intensity. The malicious SU must continuously adjust its action to maximize the amount of wasted white spaces (or tokens) thereby reducing network scalability without being detected.

B. PROPOSED RL-BASED TRUST MODEL

We propose a RL-based trust model to countermeasure the effects of random and intelligent attacks from the malicious SUs. Table 3 represents the RL model embedded in a SU cluster head CH_c.

TABLE 3. RL model embedded in a cluster head CH_c.

RL element	Representation	Description
Action	$a_{m,t} = \eta_{given}$	action represents the tokens awarded to SU m, in which $\eta_{given} = 1$ refers to tokens being awarded and $\eta_{given} = 0$ refers to tokens not being awarded.
Reward	$r_{m,t+1}(a_{m,t}) = \{0,1\}$	reward represents the utilisation level of tokens by SU m, in which 1 refers to tokens being utilised and 0 refers to tokens being wasted.

The cluster head CH_c adopts a trust model to establish trust amongst its member nodes. The cluster head CH_c awards tokens to legitimate SUs, which have comparatively high trust values or (Q-values $Q_{m,t+1}(a_{m,t})$). Note that, a cluster head CH_c increases the trust value of a SU member node if the node joins a single cluster and requests for the right amount of tokens and this is reflected by the increment of the Q-values of the legitimate SU member nodes. Once a SU m becomes a member node, it sends packets to the cluster head CH_c. However, the cluster head may not pass tokens to the

SU m if there is insufficient of white spaces. If a member node successfully sends packets to the cluster head, then it is rewarded by $r_{m,t+1} = 1$, otherwise it is rewarded by $r_{m,t+1} = 0$. The cluster head CH_c calculates the reward $r_{m,t+1}$ based on the utilisation level of the tokens (or successful packet transmissions). Next, the cluster head CH_c updates the Q-value for the state-action pair using Equation (1).

C. STATE REPRESENTATION IN THE RL-BASED ATTACK AND TRUST MODELS

The cluster head adopts a stateless RL model for the trust model, while the malicious SU node adopts a RL model with state representation for the attack model. The choice of a stateless model is because the operating environment of the cluster head does not affect the decision in the awarding of tokens. Without state representation, the action space is comparatively small compared to state-action space in models with states (which is adopted by the attack model), and so the cluster head has the capability to learn faster. Hence, the cluster head is expected to converge and perform better.

V. PERFORMANCE EVALUATION

In this section, we present the simulation setup and parameters, performance metrics, as well as results and discussions.

A. SIMULATION SETUP AND PARAMETERS

Our simulation is implemented in Microsoft Visual C++ simulator. The PUs and SUs are randomly deployed in a CRN. A SU node should only join a single cluster and request for the right number of tokens based on its hop count from its cluster head. Table 4 presents the simulation parameters and values.

TABLE 4. Simulation parameters.

Notation	Description	Value
T	Decision epochs	$\{1, 2, \ldots, 10000\}$
$\|J\|$	Number of primary users	3
$\|M\|$	Number of secondary users	$\{1, 2, \ldots, 50\}$
$a_{m_1,t}$	Probability of attack	$\{0.1, 0.2, \ldots, 0.9\}$
$a_{m_2,t}$	Intensity of attack	$\{0.1, 0.2, \ldots, 0.9\}$
$\beta_{c,t}$	Available budget	≤ 15
α	Learning rate	0.5
γ	Discount rate	0.8

For simplicity, RL and R refer to reinforcement learning and random approaches, while CH and SU refer to cluster head and malicious SU member node henceforth. We investigate four cases: a) Case I (RL-CH + RL-SU) where both cluster head and malicious SU member node use the RL approach, b) Case II (R-CH + RL-SU) where cluster head uses the random approach and malicious SU member nodes use the RL approach, c) Case III (RL-CH + R-SU) where cluster head uses the RL approach and malicious SU member nodes use the random approach, and d) Case IV (R-CH + R-SU) where both cluster head and malicious SU member node use the random approach. Investigations are conducted under varying probability of attack and intensity of attack. The probability of attack represents the likelihood of a SU node that attacks, and the intensity of attack represents the severity of an attack. For instance, an intensity of attack value of 0.1 means that 10% of the tokens (or white spaces) are wasted.

B. PERFORMANCE METRICS

The performance metrics are as follows:

- *Utilisation of white spaces ratio* is the ratio of the number of tokens needed by SU member nodes based on their hop count from their respective cluster heads to the number of tokens requested by the SU member nodes from their respective cluster heads.
- *Cluster size ratio* is the ratio of the number of SU member nodes in a cluster without attack to the number of SU member nodes in a cluster under attack.

C. RESULTS AND DISCUSSIONS

In this section, we present analysis and performance comparison. For analysis, we investigate the effects of varying probability of attack and intensity of attack on Case I (RL-CH + RL-SU). The motivation is to investigate the effects of attacks by malicious nodes against cluster heads. Results obtained in analysis are applied for performance comparison. Case I (RL-CH + RL-SU), where both malicious nodes and cluster heads have the same level of intelligence, is significant because most investigations consider malicious nodes that launch random attacks [42], [43]. Specifically, in Case I, not only is RL applied to cluster heads, but also malicious SU nodes. Our results show that, cluster heads can still perform better because the cluster head uses a stateless model that enables the cluster heads to learn faster. This serves as a proper guide for designing security schemes based on artificial intelligence. For performance comparison, we compare the performances achieved in the four cases, namely Case I (RL-CH + RL-SU), Case II (R-CH + RL-SU), Case III (RL-CH + R-SU), and Case IV (R-CH + R-SU). A cluster head aims to increase the number of nodes in a cluster (or reduce the number of clusters in a network) in order to improve network scalability. Malicious SU nodes aim to reduce the number of nodes in a cluster by joining more than a single cluster or requesting more tokens than required in order to reduce network scalability.

1) ANALYSIS ON THE EFFECTS OF VARYING PROBABILITY OF ATTACK IN CASE I (RL-CH + RL-SU)

The results obtained in this section is applied for comparison in Section V-C3.

Figure 3 shows the utilisation of white spaces (or tokens) ratio for different intensity of attack under varying probability of attack. The utilisation of white spaces generally decreases when the probability of attack increases. Besides, the utilisation of white spaces also decreases when the intensity of attack increases. This is because, when the probability of attack is high, a cluster head disqualifies and expels malicious

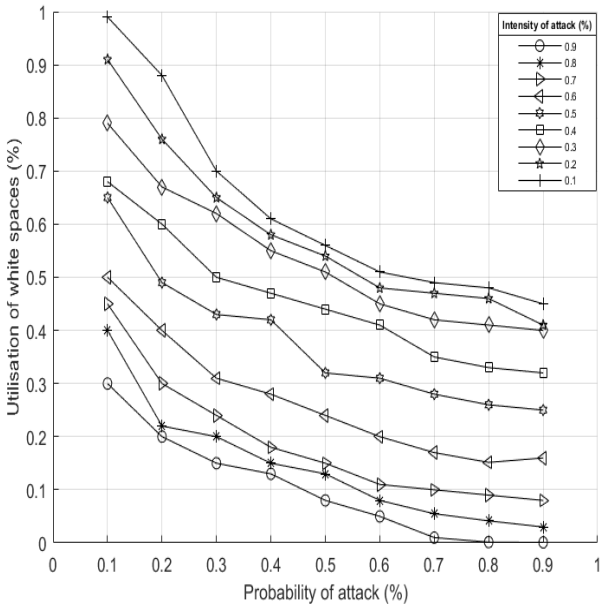

FIGURE 3. Utilisation of white spaces increases when the probability of attack decreases.

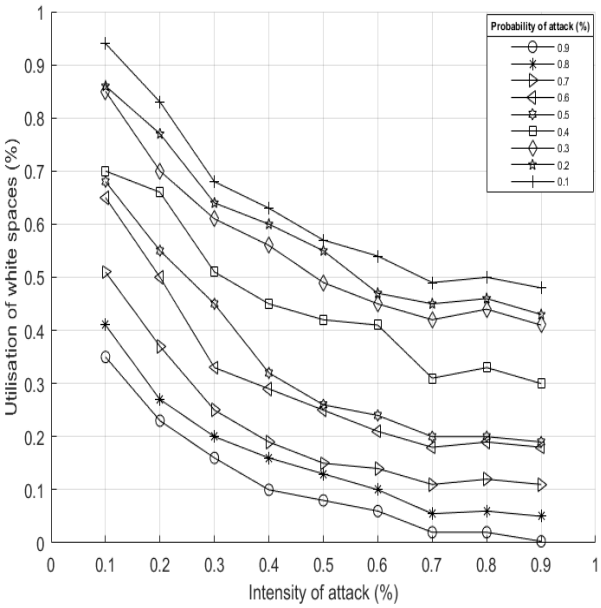

FIGURE 5. Utilisation of white spaces increases when the intensity of attack decreases.

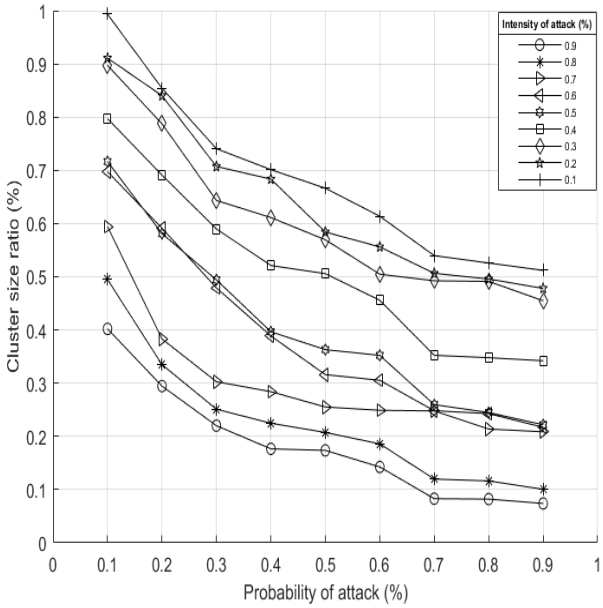

FIGURE 4. Cluster size ratio increases when the probability of attack decreases.

SUs from its cluster. In the case when a malicious SU is expelled from its cluster, it tends to adapt attack strategy so that it is not expelled from its cluster again.

Figure 4 shows the cluster size ratio for different intensity of attack under varying probability of attack. The cluster size ratio generally decreases when the probability of attack increases. Besides, the cluster size ratio also decreases when the intensity of attack increases. This means that, when the probability and intensity of attack are both low (i.e., 0.1) there is a higher cluster size ratio, which is close to a value of 1. This shows that the cluster size is not significantly affected by low probability and intensity of attack (i.e., 0.1). As the probability and intensity of attack increase, we observe a smaller cluster size ratio because some member nodes are expelled from their respective clusters for being not trustworthy. Figure 4 shows that some results are close with each other, and crossovers happen in some cases. For example, in Figure 4, there is a crossover of cluster size ratio for the result achieved by: a) probability of attack 0.2 and intensity of attack 0.5, and b) probability of attack 0.2 and intensity of attack 0.6; the amount of crossover is 0.01 or 1%. Generally speaking, the amount of each crossover is small in all cases, and crossovers occur when the probability of attack is small (i.e., less than 0.3) and large (i.e., more than 0.6), indicating that learning becomes unstable because all the possible actions become indistinguishable providing very good or very poor network performance. Similar observations are found in the literature [2]. Similar observations are also found in Figures 5, 6, 7, 8, 10 and 11.

2) ANALYSIS ON EFFECTS OF VARYING INTENSITY OF ATTACK FOR RL-CH + RL-SU

The results obtained in this section is applied for comparison in Section V-C3.

Figure 5 shows the utilisation of white spaces (or tokens) ratio for different probability of attack under varying intensity of attack. The utilisation of white spaces generally decreases when the intensity of attack increases. Besides, the utilisation of white spaces also decreases when the probability of attack increases. Figure 6 shows the cluster size ratio for different probability of attack under varying intensity of attack. The cluster size ratio decreases when the intensity of attack increases. Besides, the cluster size ratio also decreases when the probability of attack increases. The explanation on the

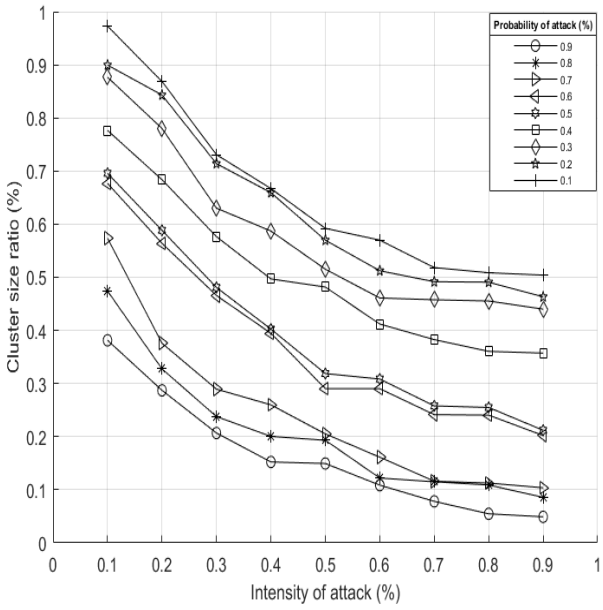

FIGURE 6. Cluster size ratio increases when the intensity of attack decreases.

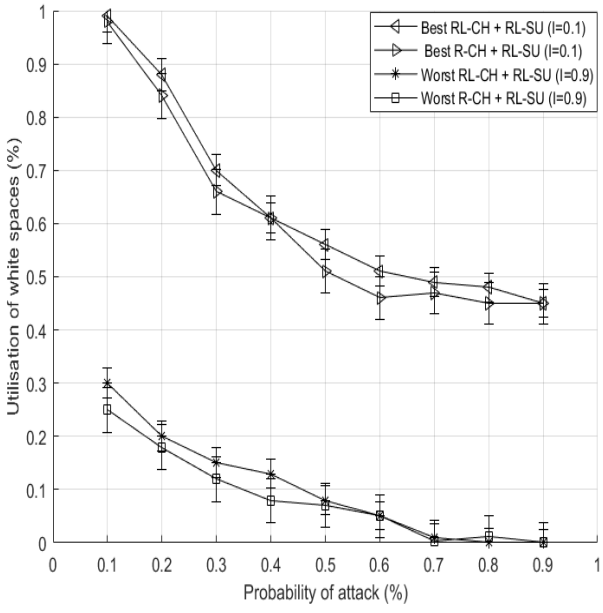

FIGURE 7. Utilisation of white spaces increases when the probability of attack decreases. Case I (RL-CH + RL-SU) achieves better performance, contributing to increased network scalability.

results in Section V-C1 applies similarly in this section.

3) PERFORMANCE COMPARISON CASE I (RL-CH + RL-SU) AND CASE II (R-CH + RL-SU)

This section presents the comparison of performance between Case I (RL-CH + RL-SU) and Case II (R-CH + RL-SU) whereby malicious SUs adopt RL-based intelligent attacks.

Figure 7 shows that the utilisation of white spaces (or tokens) ratio in Cases I and II for different intensity of

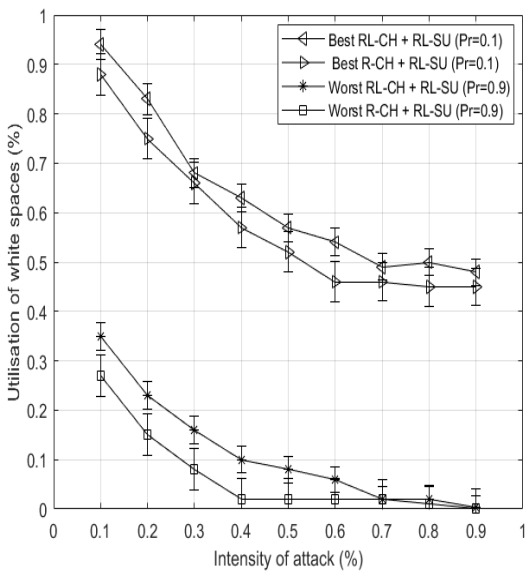

FIGURE 8. Utilisation of white spaces increases when the intensity of attack decreases. Case I (RL-CH + RL-SU) achieves better performance, contributing to increased network scalability.

attack under varying probability of attack. Case I (RL-CH + RL-SU) shows a significantly higher increase in the utilisation of white spaces as compared to Case II (R-CH + RL-SU) for varying probability of attack. There are two main reasons for this significant difference. Firstly, in Case I (RL-CH + RL-SU), the cluster head applies RL which helps the cluster head to make the right decision so that it awards tokens to SUs with comparatively higher trust values (or Q-values). Secondly, in Case I (RL-CH + RL-SU), the malicious SUs apply RL which helps them to choose their attack strategies based on the response from the cluster head. Case II (R-CH + RL-SU) achieves lower utilisation level of white spaces because the cluster head applies the random approach, therefore, the cluster head awards tokens to SUs randomly. Hence, the utilisation level of white spaces is higher in Case I (RL-CH + RL-SU) compared to the Case II (R-CH + RL-SU).

Figure 8 shows the utilisation of white spaces (or tokens) ratio under varying intensity of attack. Case I (RL-CH + RL-SU) achieves higher utilisation level of white spaces in a cluster as compared to Case II (R-CH + RL-SU) because the Q-values of Case I (RL-CH + RL-SU) increases with higher utilisation at the cluster head, allowing more SU nodes to join the cluster and utilize white spaces. Higher utilisation of white spaces increases network scalability.

Figure 9 shows the cluster size ratio in Cases I and II for different intensity of attack under varying probability of attack. Case I (RL-CH + RL-SU) achieves higher cluster size ratio compared to Case II (R-CH + RL-SU) because the cluster head applies RL which helps to make the right decision when awarding tokens to SUs with comparatively high trust value (or Q-values). With larger cluster size, there are lower number of clusters in a network, contributing to increased network scalability.

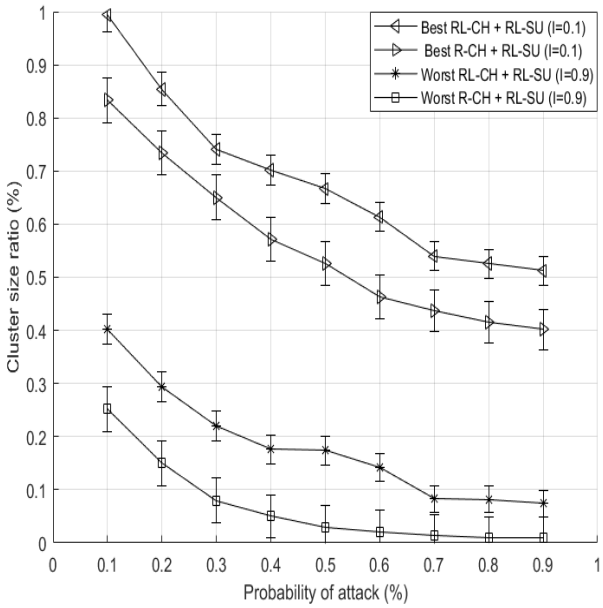

FIGURE 9. Cluster size ratio increases when the probability of attack decreases. Case I (RL-CH + RL-SU) achieves better performance, contributing to increased network scalability.

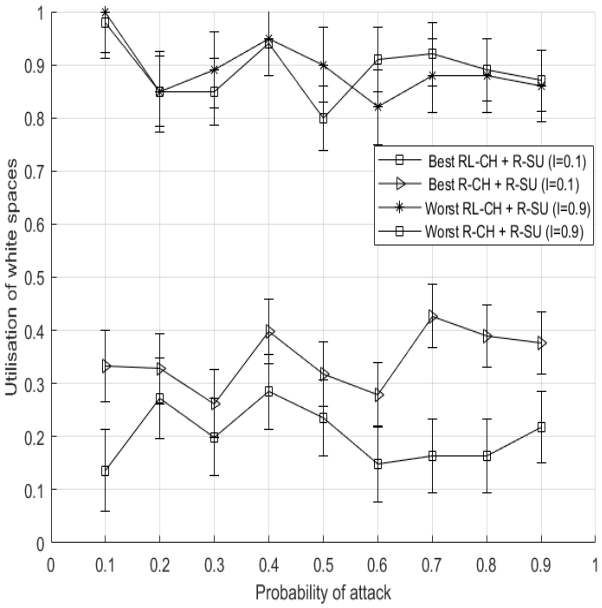

FIGURE 11. Case III (RL-CH + R-SU) achieves higher values than Case IV (R-CH + R-SU).

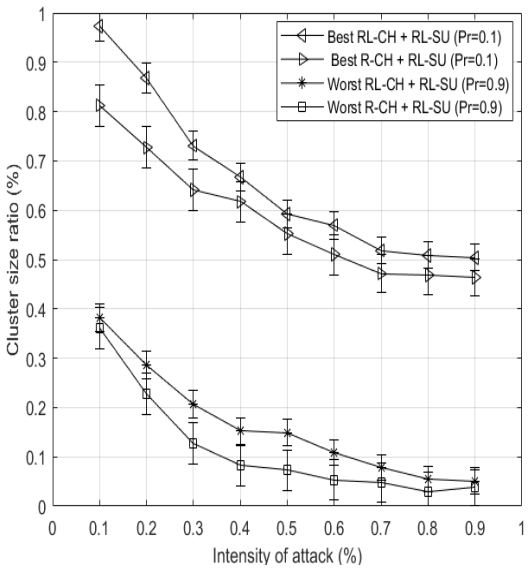

FIGURE 10. Cluster size ratio increases when the intensity of attack decreases. Case I (RL-CH + RL-SU) achieves better performance, contributing to increased network scalability.

Figure 10 shows the cluster size ratio in Cases I and II for different probability of attack under varying intensity of attack. Case I (RL-CH + RL-SU) achieves higher cluster size ratio compared to Case II (R-CH + RL-SU). The cluster size ratio under varying intensity of attack shows that the average number of nodes in a cluster for Case I (RL-CH + RL-SU) is higher because the Q-values reflect the effect of the attack against the cluster. This increases network scalability. Hence, Case I (RL-CH + RL-SU) shows better performance than Case II (R-CH + RL-SU).

4) PERFORMANCE COMPARISON CASE III (RL-CH + R-SU) AND CASE IV (R-CH + R-SU)

This section presents the comparison of performance between Case III (RL-CH + R-SU) and Case IV (R-CH + R-SU) whereby malicious SUs adopt random attacks.

Figure 11 shows that the utilisation of white spaces is unstable for different intensity of attack under varying probability of attack. Case III (RL-CH + R-SU) achieves a higher utilisation level of white spaces compared to Case IV (R-CH + R-SU). Due to the random nature of the attack, the overall performance fluctuates and this causes instability. Nevertheless, applying RL can detect the attackers easily in practice and they can be expelled from the cluster.

VI. CONCLUSION

This research presents a budget-based cluster size adjustment scheme that is applied to each cluster, in order to adjust its number of member nodes in its cluster based on the availability of white spaces to improve network scalability. An artificial intelligence approach called reinforcement learning (RL) is embedded in both cluster heads and malicious SU nodes to countermeasure the effects of attack from malicious SU nodes to form an optimal cluster size in CRN and to improve network scalability. RL is embedded in cluster heads to help make right decisions in awarding tokens to malicious SU nodes while RL is embedded in malicious SU nodes to launch intelligent attacks by observing and learning the operating environment. Case I (RL-CH + RL-SU) reflects the scenario in which RL is embedded in both cluster heads and malicious SU nodes. Although the malicious SU nodes are using RL, the cluster head is able to perform better because the cluster head adopts a stateless model while the malicious SU nodes

has a large state-action pair. Therefore, the cluster head is able to learn faster compared to the malicious SU nodes. The performance of Case I (RL-CH + RL-SU) is compared with Case II (R-CH + RL-SU) in which cluster head adopts a random approach and malicious SU nodes adopt RL, Case III (RL-CH + R-SU) in which cluster head adopts RL and malicious SU nodes adopt a random approach, and Case IV (R-CH + R-SU) in which both cluster head and malicious SU nodes adopt random approach. Simulation results show that Case I (RL-CH + RL-SU) shows stability in performance compared to the other scenarios. Hence, simulation results show that Case I (RL-CH + RL-SU) increases utilisation of white spaces and cluster size ratio compared to the other scenarios. In our future work, we aim to investigate important areas of Case I (RL-CH + RL-SU), such as implementing a mechanism that chooses the best case based on varying the RL parameters.

REFERENCES

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, Aug. 1999.

[2] K.-L. A. Yau, G. S. Poh, and P. Komisarczuk, "Security aspects in the cognition cycle of distributed cognitive radio networks: A survey from a multi-agent perspective," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 12, no. 3, pp. 157–176, Jan. 2013.

[3] Y. P. Chen, A. L. Liestman, and J. Liu, "Clustering algorithms for ad hoc wireless networks," *Ad Hoc Sensor Netw.*, vol. 28, p. 76, 2004.

[4] Y. Saleem, K.-L. A. Yau, H. Mohamad, N. Ramli, and M. H. Rehmani, "SMART: A spectrum-aware cluster-based routing scheme for distributed cognitive radio networks," *Comput. Netw.*, vol. 91, pp. 196–224, Nov. 2015.

[5] M. Ozger, F. Alagoz, and O. B. Akan, "Clustering in multi-channel cognitive radio ad hoc and sensor networks," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 156–162, Apr. 2018.

[6] F. Aftab, Z. Zhang, and A. Ahmad, "Self-organization based clustering in MANETs using zone based group mobility," *IEEE Access*, vol. 5, pp. 27464–27476, 2017.

[7] R. K. Berwer and S. Kumar, "Multi channel based clustering in cognitive radio networks," in *Proc. Int. Conf. Smart Technol. Smart Nation (SmartTechCon)*, Aug. 2017, pp. 665–670.

[8] S. Kumar and A. K. Singh, "A localized algorithm for clustering in cognitive radio networks," *J. King Saud Univ. Comput. Inf. Sci.*, to be published. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157817305256

[9] M. Ozger, E. Fadel, and O. B. Akan, "Event-to-sink spectrum-aware clustering in mobile cognitive radio sensor networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 9, pp. 2221–2233, Sep. 2016.

[10] M. Zareei, E. M. Mohamed, M. H. Anisi, C. V. Rosales, K. Tsukamoto, and M. K. Khan, "On-demand hybrid routing for cognitive radio ad-hoc network," *IEEE Access*, vol. 4, pp. 8294–8302, 2016.

[11] G. P. Joshi and S. W. Kim, "A survey on node clustering in cognitive radio wireless sensor networks," *Sensors*, vol. 16, no. 9, p. 1465, Sep. 2016.

[12] C. V. Ramamoorthy, A. Bhide, and J. Srivastava, "Reliable clustering techniques for large, mobile packet radio networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Shanghai, China, Jun. 1987, pp. 218–226.

[13] M. H. Ling, K.-L. A. Yau, J. Qadir, G. S. Poh, and Q. Ni, "Application of reinforcement learning for security enhancement in cognitive radio networks," *Appl. Soft Comput.*, vol. 37, pp. 809–829, Dec. 2015.

[14] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[15] A. Mishra and T. Alexander, "Radio communications: Components, systems, and networks," *IEEE Commun. Mag.*, vol. 55, no. 9, p. 132, Sep. 2017.

[16] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Comput. Commun.*, vol. 30, nos. 14–15, pp. 2826–2841, Oct. 2007.

[17] O. Boyinbode, H. Le, and M. Takizawa, "A survey on clustering algorithms for wireless sensor networks," *Int. J. Space Based Situated Comput.*, vol. 1, nos. 2–3, pp. 130–136, 2011.

[18] V. Katiyar, N. Chand, and S. Soni, "Clustering algorithms for heterogeneous wireless sensor network: A survey," *Int. J. Appl. Eng. Res.*, vol. 1, no. 2, p. 273, 2010.

[19] V. Kumar, S. Jain, and S. Tiwari, "Energy efficient clustering algorithms in wireless sensor networks: A survey," *Int. J. Comput. Sci. Issues*, vol. 8, no. 5, p. 259, Sep. 2011.

[20] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *sensors*, vol. 12, no. 8, pp. 11113–11153, Aug. 2012.

[21] P. Schaffer, K. Farkas, A. Horváth, T. Holczer, and L. Buttyán, "Secure and reliable clustering in wireless sensor networks: A critical survey," *Comput. Netw.*, vol. 56, no. 11, pp. 2726–2741, Jul. 2012.

[22] J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 7, no. 1, pp. 32–48, 1st Quart., 2005.

[23] L. Lazos, S. Liu, and M. Krunz, "Spectrum opportunity-based control channel assignment in cognitive radio networks," in *Proc. 6th Annu. IEEE Commun. Soc. Conf. Sensor, Commun. Netw. (SECON)*, Rome, Italy, Jun. 2009, pp. 1–9.

[24] N. Mansoor, A. K. M Muzahidul Islam, M. Zareei, and C. V. Rosales, "RARE: A spectrum aware cross-layer MAC protocol for cognitive radio ad-hoc networks," *IEEE Access*, vol. 6, pp. 22210–22227, 2018.

[25] R. Virrankoski and A. Savvidees, "TASC: Topology adaptive spatial clustering for sensor networks," in *Proc. Int. IEEE Conf Mobile Adhoc Sensor Syst. Conf.*, Dec. 2005, p. 614.

[26] W. K. Lai, C.-S. Shieh, and Y.-T. Lee, "A cluster-based routing protocol for wireless sensor networks with adjustable cluster size," in *Proc. 4th Int. IEEE Conf. Commun. Netw. (COM)*, Xian, China, Aug. 2009, pp. 1–5.

[27] H. M. N. D. Bandara and A. P. Jayasumana, "An enhanced top-down cluster and cluster tree formation algorithm for wireless sensor networks," in *Proc. Int. Conf. Ind. Inf. Syst. (ICIIS)*, Penadeniya, Sri Lanka, Aug. 2007, pp. 565–570.

[28] C.-H. Chien and M.-S. Wang, "An improving of load balancing in clustering algorithm for wireless sensor network based on distance," in *Proc. 2nd Int. Conf. Mech. Control Comput. Eng. (ICMCCE)*, Dec. 2017, pp. 165–168.

[29] Z. Javed, K.-L. A. Yau, H. Mohamad, N. Ramli, J. Qadir, and Q. Ni, "RL-budget: A learning-based cluster size adjustment scheme for cognitive radio networks," *IEEE Access*, vol. 6, pp. 1055–1072, 2018.

[30] F. B. Abdesslem, A. Ziviani, M. D. de Amorim, and P. Todorova, "Fair and flexible budget-based clustering," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dresden, Germany, Jun. 2009, pp. 1–5.

[31] R. Krishnan and D. Starobinski, "Efficient clustering algorithms for self-organizing wireless sensor networks," *Ad Hoc Netw.*, vol. 4, no. 1, pp. 36–59, Jan. 2006.

[32] L. Jianwu, F. Zebing, F. Zhiyong, and Z. Ping, "A survey of security issues in cognitive radio networks," *China Commun.*, vol. 12, no. 3, pp. 132–150, Mar. 2015.

[33] N. Armi, W. Gharibi, W. Z. Khan, H. Zangoti, S. Rizvi, and C. Wael, "Error detection of malicious user attack in cognitive radio networks," in *Proc. Int. IEEE Conf. Radar, Antenna, Microw., Electron., Telecommun. (ICRAMET)*, Oct. 2017, pp. 89–92.

[34] S. R. Sabuj, M. Hamamura, and S. Kuwamura, "Detection of intelligent malicious user in cognitive radio network by using friend or foe (FoF) detection technique," in *Proc. Int. IEEE Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2015, pp. 155–160.

[35] E. Soltanmohammadi, M. Orooji, and M. Naraghi-Pour, "Decentralized hypothesis testing in wireless sensor networks in the presence of misbehaving nodes," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 205–215, Jan. 2013.

[36] A. Velayudham, G. V. S. Gohila, B. Hariharan, and M. M. R. Selvi, "A novel coalition game theory based resource allocation and selfish attack avoidance in cognitive radio ad-hoc networks," *J. Theor. Appl. Inf. Technol.*, vol. 64, no. 1, pp. 180–189, 2014.

[37] R. A. Priyadharshini and K. U. Haimavathi, "Detection of attacks and countermeasures in cognitive radio network," in *Proc. Int. IEEE Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2016, pp. 1102–1106.

[38] T. C. Clancy and N. Goergen, "Security in cognitive radio networks: Threats and mitigation," in *Proc. 3rd Int. IEEE Conf. Cogn. Radio Oriented Wireless Netw. Commun. (CrownCom)*, May 2008, pp. 1–8.

[39] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "Mitigating byzantine attacks in ad hoc wireless networks," Dept. Comput. Sci., Johns Hopkins Univ., Tech. Rep. Version 1, 2004, p. 16.
[40] N. Vučević, I. F. Akyildiz, and J, P.-Romero, "Dynamic cooperator selection in cognitive radio networks," *Ad Hoc Netw.*, vol. 10, no. 5, pp. 789–802, Jul. 2012.
[41] M. H. Ling, K.-L. A. Yau, and G. S. Poh, "Trust and reputation management in cognitive radio networks: A survey," *Secur. Commun. Netw.*, vol. 7, no. 11, pp. 2160–2179, Nov. 2014.
[42] O. Mistry, A. Gürsel, and S. Sen, "Comparing trust mechanisms for monitoring aggregator nodes in sensor networks," in *Proc. 8th Int. Conf. Auto. Agents Multiagent Syst.*, vol. 2, May 2009, pp. 985–992.
[43] B. Wang, Y. Wu, K. J. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE J. Sel. Areas Commun. Syst.*, vol. 29, no. 4, pp. 877–889, Apr. 2011.
[44] K. Maneenil and W. Usaha, "Preventing malicious nodes in ad hoc networks using reinforcement learning," in *Proc. 2nd Int. IEEE Symp. Wireless Commun. Syst.*, Sep. 2005, pp. 289–292.
[45] B. F. Lo, "A survey of common control channel design in cognitive radio networks," *Phys. Commun.*, vol. 4, no. 1, pp. 26–39, Mar. 2011.
[46] R. D. Kadu and P. P. Karde, "Improving performance of CSS cognitive radio networks under jamming attack," in *Proc. 2nd Int. IEEE Conf. Commun. Syst., Comput. IT Appl. (CSCITA)*, Apr. 2017, pp. 213–217.
[47] S. D. Babar, N. R. Prasad, and R. Prasad, "Countermeasure for intelligent cluster-head jamming attack in wireless sensor network," in *Proc. Int. IEEE Conf. Privacy Sec. Mobile Syst. (PRISMS)*, Jun. 2013, pp. 1–8.
[48] K. Ezirim, E. Troja, and S. Sengupta, "Sustenance against RL-based Sybil attacks in cognitive radio networks using dynamic reputation system," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Nov. 2013, pp. 1789–1794.
[49] A. S. Cacciapuoti, M. Caleffi, and L. Paura, "Reactive routing for mobile cognitive radio ad hoc networks," *Ad Hoc Netw.*, vol. 10, no. 5, pp. 803–815, Jul. 2012.
[50] S. Bayhan and F. Alagöz, "A Markovian approach for best-fit channel selection in cognitive radio networks," *Ad Hoc Netw.*, vol. 12, pp. 165–177, Jan. 2014.
[51] M. H. Rehmani, A. C. Viana, H. Khalife, and S. Fdida, "Surf: A distributed channel selection strategy for data dissemination in multi-hop cognitive radio networks," *Comput. Commun.*, vol. 36, nos. 10–11, pp. 1172–1185, Jun. 2013.

ANITA LATSMI MANOHAR received the B.Sc. degree (Hons.) in computer science from Sunway University, Malaysia, under the dual degree program of Sunway University and Lancaster University, U.K., in 2015. She is currently pursuing the M.Sc. degree in computer science with Sunway University, under the dual degree program of Sunway University and Lancaster University. Her research interests are wireless networks, particularly, applied reinforcement learning, and cognitive radio networks.

KOK-LIM ALVIN YAU received the B.Eng. degree (Hons.) in electrical and electronics engineering from the Universiti Teknologi Petronas, Malaysia, in 2005, the M.Sc. degree in electrical engineering from the National University of Singapore, in 2007, and the Ph.D. degree in network engineering from the Victoria University of Wellington, New Zealand, in 2010. He received the 2007 Professional Engineer Board of Singapore Gold Medal for being the Best Graduate of the M.Sc. degree from 2006 to 2007. He is currently an Associate Professor with Sunway University, Malaysia. He researches, lectures, and consults in cognitive radio, wireless networking, and applied artificial intelligence.

MEE HONG LING received the Bachelor's degree (Hons.) in computer and mathematics from Oxford Brookes University, U.K., and the master's degree in data engineering (computer science) from Keele University, U.K. She is currently pursuing the Ph.D. degree. She is a Lecturer with the Department of Computing and Information Systems, Sunway University. Her research interests are in the areas of security, cognitive radio networks, and applied reinforcement learning.

SULEMAN KHAN received the Ph.D. degree (Hons.) from the Faculty of Computer Science and Information Technology, University of Malaya, Malaysia, in 2017. He is a Faculty Member with the School of Information Technology, Monash University Malaysia Campus. He has published more than 45 high-impact research articles in reputed international journals, including the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, *ACM Computing Surveys*, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He has published in the 2018 Local Computer Networks Conference. His research areas include, but are not limited to, network forensics, software-defined networks, the Internet of Things, cloud computing, and vehicular communications.